PCI Code and ID Assignment Specification Revision 1.11

24 Jan 2019

Revision	Revision History	Date
1.0	Initial release.	9/9/2010
1.1	Incorporated approved ECNs.	3/15/2012
1.2	Incorporated ECN for Accelerator Class code, added PI for xHCI.	3/15/2012
1.3	Updated section 1.2, *Base Class 01h,* Sub-class 00h by adding Programming Interfaces 11h, 12h, 13h, and 21h. Added Notes 3, 4, and 5.	9/4/2012
1.4	Updated Section 1.2, *Base Class 01h*, to add Sub-class 09h. Updated Section 1.9, *Base Class 08h* to add Root Complex Event Collector, Sub-class 07h Updated Section 1 and added Section 1.20, to define *Base Class 13h*. Updated Chapter 3 to define Extended Capability IDs 001Dh through 0022h. Reformatted Notes in Sections 1.2 and 1.7 through 1.10. Updated references to NVM Express in Section 1.9, *Base Class 08h*	8/29/2013
1.5	Updated Section 1.2, to clarify SOP entries in *Base Class 01h*, add proper reference to NVMHCI, update UFS entries, and address other minor editorial issues. Updated Section 3, Extended Capability ID descriptions 19h, 1Ch, 1Fh.	3/6/2014
1.6	Updated Section 1.3, Class 02h, to add Sub-Class 08h. Updated Section 1.14, Base Class 0Dh, to add Sub-Classes 40h and 41h. Updated Section 2 to add Capability ID 14h.	12/9/2014
1.7	Added Designated Vendor-Specific Extended Capability ID. Updated/Modified Section 1.5, Base Class 04h, for Multimedia devices to accurately reflect use of this class for High Definition Audio (HD-A).	8/13/2015
1.8	Small edits. Added Extended Capability IDs for: • VF Resizable BAR • Data Link Feature • Physical Layer 16.0 GT/s • Lane Margining at the Receiver	9/1/2016
1.9	Added the Hierarchy ID Extended Capability ID. Added the Flattening Portal Bridge Capability ID. Added Class/Sub-Class/PI for I3C Host Controller.	5/18/2017
1.10	Added NPEM New legal boilerplate language (p.3)	11/8/2017
1.11	Added: Physical Layer 32.0 GT/s Extended Capability ID Alternate Protocol Extended Capability ID System Firmware Intermediary Extended Capability ID Fixed errata re: name of “TPH Requester” Extended Capability Added Programming Interface for NVM Express (NVMe) administrative controller and related text changes Assorted editorial corrections and enhancements	1/24/2019

PCI-SIG® disclaims all warranties and liability for the use of this document and the information contained herein and assumes no responsibility for any errors that may appear in this document, nor does PCI-SIG make a commitment to update the information contained herein.

Contact the PCI-SIG office to obtain the latest revision of this specification.

Questions regarding the PCI Code and ID Assignment Specification or membership in PCI-SIG may be forwarded to:

Membership Services
www.pcisig.com
E-mail: administration@pcisig.com
Phone: 503-619-0569
Fax: 503-644-6708

Technical Support
techsupp@pcisig.com

DISCLAIMER

This specification is the sole property of PCI-SIG® and provided under a click through license through its website, www.pci-sig.com. PCI-SIG disclaims all warranties and liability for the use of this document and the information contained herein and assumes no responsibility for any errors that may appear in this document, nor does PCI-SIG make a commitment to update the information contained herein.

This PCI Specification is provided "as is" without any warranties of any kind, including any warranty of merchantability, non-infringement, fitness for any particular purpose, or any warranty otherwise arising out of any proposal, specification, or sample. PCI-SIG disclaims all liability for infringement of proprietary rights, relating to use of information in this specification. This document itself may not be modified in any way, including by removing the copyright notice or references to PCI-SIG. No license, express or implied, by estoppel or otherwise, to any intellectual property rights is granted herein. PCI, PCI Express, PCIe, and PCI-SIG are trademarks or registered trademarks of PCI-SIG. All other product names are trademarks, registered trademarks, or servicemarks of their respective owners.

Copyright © PCI-SIG 2019. All Rights Reserved.

Contents

OBJECTIVE OF THE SPECIFICATION **6**
REFERENCE DOCUMENTS **6**
DOCUMENTATION CONVENTIONS **6**
TERMS AND ACRONYMS **7**
1. CLASS CODES **8**
1.1. BASE CLASS 00H 9
1.2. BASE CLASS 01H 9
1.3. BASE CLASS 02H 11
1.4. BASE CLASS 03H 11
1.5. BASE CLASS 04H 12
1.6. BASE CLASS 05H 12
1.7. BASE CLASS 06H 13
1.8. BASE CLASS 07H 14
1.9. BASE CLASS 08H 15
1.10. BASE CLASS 09H 16
1.11. BASE CLASS 0AH 16
1.12. BASE CLASS 0BH 17
1.13. BASE CLASS 0CH 18
1.14. BASE CLASS 0DH 19
1.15. BASE CLASS 0EH 19
1.16. BASE CLASS 0FH 19
1.17. BASE CLASS 10H 20
1.18. BASE CLASS 11H 20
1.19. BASE CLASS 12H 20
1.20. BASE CLASS 13H 21
2. CAPABILITY IDS **22**
3. EXTENDED CAPABILITY IDS **24**

Tables

TABLE 2-1: CAPABILITY IDS 22
TABLE 3-1: EXTENDED CAPABILITY IDS 24

Objective of the Specification

This specification contains the Class Code and Capability ID descriptions originally contained the *PCI Local Bus Specification,* bringing them into a standalone document that is easier to reference and maintain. This specification also consolidates Extended Capability ID assignments from the *PCI Express Base Specification* and various other PCI specifications.

Reference Documents

PCI Express Base Specification

PCI Local Bus Specification

PCI-X Protocol Addendum to the PCI Local Bus Specification

Documentation Conventions

Capitalization

Some terms are capitalized to distinguish their definition in the context of this document from their common English meaning. Words not capitalized have their common English meaning. When terms such as "memory write" or "memory read" appear completely in lower case, they include all transactions of that type.

Register names and the names of fields and bits in registers and headers are presented with the first letter capitalized and the remainder in lower case.

Numbers and Number Bases

Hexadecimal numbers are written with a lower case "h" suffix, e.g., FFFh and 80h. Hexadecimal numbers larger than four digits are represented with a space dividing each group of four digits, as in 1E FFFF FFFFh. Binary numbers are written with a lower case "b" suffix, e.g., 1001b and 10b. Binary numbers larger than four digits are written with a space dividing each group of four digits, as in 1000 0101 0010b.

All other numbers are decimal.

Terms and Acronyms

Base Class	The upper byte of a Class Code, which broadly classifies the type of functionality that the device Function provides.
Capability ID	An eight-bit value that identifies the type and format of a PCI-Compatible Capability structure. See the *PCI Local Bus Specification*.
Class Code	A three-byte field in a Function's Configuration Space header that identifies the generic functionality of the Function, and in some cases, a specific Programming Interface. See the *PCI Local Bus Specification*.
Extended Capability ID	A sixteen-bit value that identifies the type and format of an Extended Capability structure. See the *PCI Express Base Specification*.
Programming Interface	The lower byte of a Class Code, which identifies the specific register-level interface (if any) of a device Function, so that device-independent software can interact with the device.
Sub-Class	The middle byte of a Class Code, which more specifically identifies the type of functionality that the device Function provides.
Vendor-Specific	Behavior defined by the manufacturer identified by the Vendor ID field in the PCI Capability Header (Configuration Space offset 00h).

1

1. Class Codes[1]

This chapter describes the current Class Code encodings. This list may be enhanced at any time. The PCI-SIG web site contains the latest version of this specification. Companies wishing to define a new encoding should contact the PCI-SIG. All unspecified values are reserved for PCI-SIG assignment.

Base Class	Meaning
00h	Device was built before Class Code definitions were finalized
01h	Mass storage controller
02h	Network controller
03h	Display controller
04h	Multimedia device
05h	Memory controller
06h	Bridge device
07h	Simple communication controllers
08h	Base system peripherals
09h	Input devices
0Ah	Docking stations
0Bh	Processors
0Ch	Serial bus controllers
0Dh	Wireless controller
0Eh	Intelligent I/O controllers
0Fh	Satellite communication controllers
10h	Encryption/Decryption controllers
11h	Data acquisition and signal processing controllers
12h	Processing accelerators
13h	Non-Essential Instrumentation
14h - FEh	Reserved
FFh	Device does not fit in any defined classes

[1] The content of this chapter was originally in Appendix D of the *PCI Local Bus Specification, Revision 3.0*.

1.1. Base Class 00h

This base class is defined to provide backward compatibility for devices that were built before the Class Code field was defined. No new devices should use this value and existing devices should switch to a more appropriate value if possible.

For class codes with this base class value, there are two defined values for the remaining fields as shown in the table below. All other values are reserved.

Base Class	Sub-Class	Programming Interface	Meaning
00h	00h	00h	All currently implemented devices except VGA-compatible devices
	01h	00h	VGA-compatible device

1.2. Base Class 01h

This base class is defined for all types of mass storage controllers. Several sub-class values are defined.

Base Class	Sub-Class	Programming Interface	Meaning
01h	00h	00h	SCSI controller - vendor-specific interface
		11h	SCSI storage device (e.g., hard disk drive (HDD), solid state drive (SSD), or RAID controller) - SCSI over PCI Express (SOP) target port using PCI Express Queuing Interface (PQI) (see Notes 3 and 4)
		12h	SCSI controller (i.e., host bus adapter) - SCSI over PCI Express (SOP) target port using PCI Express Queuing Interface (PQI) (see Notes 3 and 4)
		13h	SCSI storage device and SCSI controller - SCSI over PCI Express (SOP) target port using PCI Express Queuing Interface (PQI) (see Notes 3 and 4)
		21h	SCSI storage device - SCSI over PCI Express (SOP) target port using the queueing interface portion of the NVM Express interface (see Notes 3 and 6)
	01h	xxh	IDE controller (see Note 1)
	02h	00h	Floppy disk controller - vendor-specific interface
	03h	00h	IPI bus controller - vendor-specific interface
	04h	00h	RAID controller - vendor-specific interface
	Table continues on the following page		

Base Class	Sub-Class	Programming Interface	Meaning
01h	05h	20h	ATA controller with ADMA interface - single stepping (see Note 2)
		30h	ATA controller with ADMA interface - continuous operation (see Note 2)
	06h	00h	Serial ATA controller - vendor-specific interface
		01h	Serial ATA controller - AHCI interface (see note 7)
		02h	Serial Storage Bus Interface
	07h	00h	Serial Attached SCSI (SAS) controller - vendor-specific interface
		01h	Obsolete
	08h	00h	Non-volatile memory subsystem - vendor-specific interface
		01h	Non-volatile memory subsystem - NVMHCI interface (see note 8)
		02h	NVM Express (NVMe) I/O controller (see Note 6)
		03h	NVM Express (NVMe) administrative controller (see Note 6)
	09h	00h	Universal Flash Storage (UFS) controller - vendor-specific interface
		01h	Universal Flash Storage (UFS) controller - Universal Flash Storage Host Controller Interface (UFSHCI) (see Note 5)
	80h	00h	Other mass storage controller - vendor-specific interface

Notes:

1. Register interface conforms to the PCI Compatibility and PCI-Native Mode Bus interface defined in ANSI INCITS370-2004: ATA Host Adapters Standard (see http://www.incits.org and http://www.t13.org).
2. Register interface conforms to the ADMA interface defined in ANSI INCITS 370-2004: ATA Host Adapters Standard (see http://www.incits.org and http://www.t13.org).
3. Conforms to the SCSI over PCI Express (SOP) standard (ISO/IEC 14776-271) (see http://www.incits.org and http://www.t10.org).
4. Conforms to PCI Express Queuing Interface (PQI) standard (ISO/IEC 14776-171) (see http://www.incits.org and http://www.t10.org).

5 Conforms to JESD223a (see http://www.jedec.org/standards-documents/docs/jesd223a).
6 Conforms to the NVM Express Specification (see http://www.nvmexpress.org).
7 Conforms to the AHCI Specification (see http://www.intel.com).
8 Conforms to the NVMHCI Specification (see http://www.nvmexpress.org).

1.3. Base Class 02h

This base class is defined for all types of network controllers. Several sub-class values are defined.

Base Class	Sub-Class	Programming Interface	Meaning
02h	00h	00h	Ethernet controller
	01h	00h	Token Ring controller
	02h	00h	FDDI controller
	03h	00h	ATM controller
	04h	00h	ISDN controller
	05h	00h	WorldFip controller
	06h	xxh (see Note 1 below)	PICMG 2.14 Multi Computing
	07h	00h	InfiniBand* Controller
	08h	00h	Host fabric controller – vendor-specific
	80h	00h	Other network controller

Notes:
1. For information on the use of this field see the PICMG 2.14 Multi Computing Specification (http://www.picmg.com).

1.4. Base Class 03h

This base class is defined for all types of display controllers. For VGA devices (Sub-Class 00h), the Programming Interface byte is divided into a bit field that identifies additional video controller compatibilities. A device can support multiple interfaces by using the bit map to indicate which interfaces are supported. For the XGA devices (Sub-Class 01h), only the standard XGA interface is defined. Sub-Class 02h is for controllers that have hardware support for 3D operations and are not VGA compatible.

Base Class	Sub-Class	Programming Interface	Meaning
03h	00h	0000 0000b	VGA-compatible controller. Memory addresses 0A 0000h through 0B FFFFh. I/O addresses 3B0h to 3BBh and 3C0h to 3DFh and all aliases of these addresses.
		0000 0001b	8514-compatible controller. I/O addresses 2E8h and its aliases, 2EAh-2EFh
	01h	00h	XGA controller
	02h	00h	3D controller
	80h	00h	Other display controller

1.5. Base Class 04h

This base class is defined for all types of multimedia devices. Several sub-class values are defined.

Base Class	Sub-Class	Programming Interface	Meaning
04h	00h	00h	Video device – vendor specific interface
	01h	00h	Audio device – vendor specific interface
	02h	00h	Computer telephony device – vendor specific interface
	03h	00h	High Definition Audio (HD-A) 1.0 compatible (see Note 1)
		80h	High Definition Audio (HD-A) 1.0 compatible (see Note 1) with additional vendor specific extensions
	80h	00h	Other multimedia device – vendor specific interface

Notes:

1. The High Definition Audio Specification is available here: http://www.intel.com/content/www/us/en/standards/standards-high-def-audio-specs-general-technology.html

1.6. Base Class 05h

This base class is defined for all types of memory controllers. Several sub-class values are defined. There are no Programming Interfaces defined.

Base Class	Sub-Class	Programming Interface	Meaning
05h	00h	00h	RAM
	01h	00h	Flash
	80h	00h	Other memory controller

1.7. Base Class 06h

This base class is defined for all types of bridge devices. A PCI bridge is any PCI device that maps PCI resources (memory or I/O) from one side of the device to the other. Several sub-class values are defined.

Base Class	Sub-Class	Programming Interface	Meaning
06h	00h	00h	Host bridge
	01h	00h	ISA bridge
	02h	00h	EISA bridge
	03h	00h	MCA bridge
	04h	00h	PCI-to-PCI bridge
		01h	Subtractive Decode PCI-to-PCI bridge. This interface code identifies the PCI-to-PCI bridge as a device that supports subtractive decoding in addition to all the currently defined functions of a PCI-to-PCI bridge.
	05h	00h	PCMCIA bridge
	06h	00h	NuBus bridge
	07h	00h	CardBus bridge
	08h	xxh	RACEway bridge (see Note 1 below)
	09h	40h	Semi-transparent PCI-to-PCI bridge with the primary PCI bus side facing the system host processor
		80h	Semi-transparent PCI-to-PCI bridge with the secondary PCI bus side facing the system host processor
	0Ah	00h	InfiniBand-to-PCI host bridge
	0Bh	00h	Advanced Switching to PCI host bridge–Custom Interface
		01h	Advanced Switching to PCI host bridge– ASI-SIG Defined Portal Interface
	80h	00h	Other bridge device

Notes:

1. RACEway is an ANSI standard (ANSI/VITA 5-1994) switching fabric. For the Programming Interface bits, [7:1] are reserved, read-only, and return zeros. Bit 0 defines the operation mode and is read-only:
 0 - Transparent mode
 1 - End-point mode

1.8. Base Class 07h

This base class is defined for all types of simple communications controllers. Several sub-class values are defined, some of these having specific well-known Programming Interfaces.

Base Class	Sub-Class	Programming Interface	Meaning
07h	00h	00h	Generic XT-compatible serial controller
		01h	16450-compatible serial controller
		02h	16550-compatible serial controller
		03h	16650-compatible serial controller
		04h	16750-compatible serial controller
		05h	16850-compatible serial controller
		06h	16950-compatible serial controller
	01h	00h	Parallel port
		01h	Bi-directional parallel port
		02h	ECP 1.X compliant parallel port
		03h	IEEE1284 controller
		FEh	IEEE1284 target device (not a controller)
	02h	00h	Multiport serial controller
	03h	00h	Generic modem
		01h	Hayes compatible modem, 16450-compatible interface (see Note 1 below)
		02h	Hayes compatible modem, 16550-compatible interface (see Note 1 below)
		03h	Hayes compatible modem, 16650-compatible interface (see Note 1 below)
		04h	Hayes compatible modem, 16750-compatible interface (see Note 1 below)
	04h	00h	GPIB (IEEE 488.1/2) controller
	05h	00h	Smart Card
	80h	00h	Other communications device

Notes:

1. For Hayes-compatible modems, the first Base Address register (at offset 10h) maps the appropriate compatible (i.e., 16450, 16550, etc.) register set for the serial controller at the beginning of the mapped space. Note that these registers can be either memory or I/O mapped depending what kind of BAR is used.

1.9. Base Class 08h

This base class is defined for all types of generic system peripherals. Several sub-class values are defined, most of these having a specific well-known Programming Interface.

Base Class	Sub-Class	Programming Interface	Meaning
08h	00h	00h	Generic 8259 PIC
		01h	ISA PIC
		02h	EISA PIC
		10h	I/O APIC interrupt controller (see Note 1 below)
		20h	I/O(x) APIC interrupt controller
	01h	00h	Generic 8237 DMA controller
		01h	ISA DMA controller
		02h	EISA DMA controller
	02h	00h	Generic 8254 system timer
		01h	ISA system timer
		02h	EISA system timers (two timers)
		03h	High Performance Event Timer
	03h	00h	Generic RTC controller
		01h	ISA RTC controller
	04h	00h	Generic PCI Hot-Plug controller
	05h	00h	SD Host controller
	06h	00h	IOMMU
	07h	00h	Root Complex Event Collector (see Note 2 below)
	80h	00h	Other system peripheral

Notes:

1 For I/O APIC Interrupt Controller, the Base Address register at offset 10h is used to request a minimum of 32 bytes of non-prefetchable memory. Two registers within that space are located at Base+00h (I/O Select register) and Base+10h (I/O Window register).

2 Some versions of the *PCI Express Base Specification* defined Root Complex Event Collectors to use Sub-class 06h. Implementations are permitted to use Sub-class 06h for this purpose, but this practice is strongly discouraged. The Device/Port Type field value can be used to accurately identify all Root Complex Event Collectors.

1.10. Base Class 09h

This base class is defined for all types of input devices. Several sub-class values are defined. A Programming Interface is defined for gameport controllers.

Base Class	Sub-Class	Programming Interface	Meaning
09h	00h	00h	Keyboard controller
	01h	00h	Digitizer (pen)
	02h	00h	Mouse controller
	03h	00h	Scanner controller
	04h	00h	Gameport controller (generic)
		10h	Gameport controller (see Note 1 below)
	80h	00h	Other input controller

Notes:

1 A gameport controller with a Programming Interface == 10h indicates that any Base Address registers in this Function that request/assign I/O address space, the registers in that I/O space conform to the standard "legacy" game ports. The byte at offset 00h in an I/O region behaves as a legacy gameport interface where reads to the byte return joystick/gamepad information, and writes to the byte start the RC timer. The byte at offset 01h is an alias of the byte at offset 00h. All other bytes in an I/O region are unspecified and can be used in vendor unique ways.

1.11. Base Class 0Ah

This base class is defined for all types of docking stations. No specific Programming Interfaces are defined.

Base Class	Sub-Class	Programming Interface	Meaning
0Ah	00h	00h	Generic docking station
	80h	00h	Other type of docking station

1.12. Base Class 0Bh

This base class is defined for all types of processors. Several sub-class values are defined corresponding to different processor types or instruction sets. There are no specific Programming Interfaces defined.

Base Class	Sub-Class	Programming Interface	Meaning
0Bh	00h	00h	386
	01h	00h	486
	02h	00h	Pentium
	10h	00h	Alpha
	20h	00h	PowerPC
	30h	00h	MIPS
	40h	00h	Co-processor
	80h	00h	Other processors

1.13. Base Class 0Ch

This base class is defined for all types of serial bus controllers. Several sub-class values are defined.

Base Class	Sub-Class	Programming Interface	Meaning
0Ch	00	00h	IEEE 1394 (FireWire)
		10h	IEEE 1394 following the 1394 OpenHCI specification
	01h	00h	ACCESS.bus
	02h	00h	SSA
	03h	00h	Universal Serial Bus (USB) following the Universal Host Controller Specification
		10h	Universal Serial Bus (USB) following the Open Host Controller Specification
		20h	USB 2 host controller following the Intel Enhanced Host Controller Interface Specification
		30h	Universal Serial Bus (USB) Host Controller following the Intel eXtensible Host Controller Interface (xHCI) Specification
		80h	Universal Serial Bus with no specific Programming Interface
		FEh	USB device (not host controller)
	04h	00h	Fibre Channel
	05h	00h	SMBus (System Management Bus)
	06h	00h	InfiniBand–This sub-class is deprecated. New InfiniBand adapters should use the base class and sub-class defined in Section 1.3 .
	07h (see Note 1 below)	00h	IPMI SMIC Interface
		01h	IPMI Keyboard Controller Style Interface
		02h	IPMI Block Transfer Interface
	08h (see Note 2 below)	00h	SERCOS Interface Standard (IEC 61491)
	09h	00h	CANbus
	0Ah (see Note 3 below)	00h	MIPI I3C℠ Host Controller Interface
	80h	00h	Other Serial Bus Controllers

Notes:

1. The register interface definitions for the Intelligent Platform Management Interface (Sub-Class 07h) are in the IPMI specification.
2. There is no register level definition for the SERCOS Interface standard. For more information see IEC 61491.
3. The MIPI I3C Host Controller Interface specification is(upon publication) available at http://software.mipi.org.

1.14. Base Class 0Dh

This base class is defined for all types of wireless controllers. Several sub-class values are defined.

Base Class	Sub-Class	Programming Interface	Meaning
0Dh	00	00h	iRDA compatible controller
	01h	00h	Consumer IR controller
		10h	UWB Radio controller
	10h	00h	RF controller
	11h	00h	Bluetooth
	12h	00h	Broadband
	20h	00h	Ethernet (802.11a – 5 GHz)
	21h	00h	Ethernet (802.11b – 2.4 GHz)
	40h	00h	Cellular controller/modem
	41h	00h	Cellular controller/modem plus Ethernet (802.11)
	80h	00h	Other type of wireless controller

1.15. Base Class 0Eh

This base class is defined for intelligent I/O controllers. The primary characteristic of this base class is that the I/O function provided follows some sort of generic definition for an I/O controller.

Base Class	Sub-Class	Programming Interface	Meaning
0Eh	00	xxh	*Intelligent I/O (I2O) Architecture Specification 1.0*
		00h	Message FIFO at offset 040h

1.16. Base Class 0Fh

This base class is defined for satellite communication controllers. Several sub-class values are defined. There are no Programming Interfaces defined.

Base Class	Sub-Class	Programming Interface	Meaning
0Fh	01h	00h	TV
	02h	00h	Audio
	03h	00h	Voice
	04h	00h	Data
	80h	00h	Other satellite communication controller

1.17. Base Class 10h

This base class is defined for all types of encryption and decryption controllers. Several sub-class values are defined. There are no Programming Interfaces defined.

Base Class	Sub-Class	Programming Interface	Meaning
10h	00h	00h	Network and computing encryption and decryption controller
	10h	00h	Entertainment encryption and decryption controller
	80h	00h	Other encryption and decryption controller

1.18. Base Class 11h

This base class is defined for all types of data acquisition and signal processing controllers. Several sub-class values are defined. There are no Programming Interfaces defined.

Base Class	Sub-Class	Programming Interface	Meaning
11h	00h	00h	DPIO modules
	01h	00h	Performance counters
	10h	00h	Communications synchronization plus time and frequency test/measurement
	20h	00h	Management card
	80h	00h	Other data acquisition/signal processing controllers

1.19. Base Class 12h

This base class is defined for processing accelerators. No sub-classes or Programming Interfaces are defined.

Base Class	Sub-Class	Programming Interface	Meaning
12h	00h	00h	Processing Accelerator – vendor-specific interface

1.20. Base Class 13h

This base class is defined for Functions that provide component/platform instrumentation capabilities not essential to normal run-time operation. Examples include instrumentation or debug capabilities used in development, or by authorized users.

It is intended that a system might implement differentiated policies for Functions with this base class, for example, a policy of silently ignoring cases where no device driver matches the Function (vs. the typical default of notifying the user).

Base Class	Sub-Class	Programming Interface	Meaning
13h	00h	00h	Non-Essential Instrumentation Function – Vendor-specific interface

2

2. Capability IDs[2]

This chapter describes the current PCI-Compatible Capability IDs. Each Capability structure must have a Capability ID assigned by the PCI-SIG. Companies wishing to define a new encoding should contact the PCI-SIG. All unspecified values are reserved for PCI-SIG assignment.

Table 2-1: Capability IDs

ID	Capability
00h	Null Capability – This capability contains no registers other than those described below. It may be present in any Function. Functions may contain multiple instances of this capability. The Null Capability is 16 bits and contains an 8-bit Capability ID followed by an 8-bit Next Capability Pointer.
01h	PCI Power Management Interface – This Capability structure provides a standard interface to control power management features in a device Function. It is fully documented in the *PCI Bus Power Management Interface Specification*.
02h	AGP – This Capability structure identifies a controller that is capable of using Accelerated Graphics Port features. Full documentation can be found in the *Accelerated Graphics Port Interface Specification.*
03h	VPD – This Capability structure identifies a device Function that supports Vital Product Data. Full documentation of this feature can be found in the *PCI Local Bus Specification*.
04h	Slot Identification – This Capability structure identifies a bridge that provides external expansion capabilities. Full documentation of this feature can be found in the *PCI-to-PCI Bridge Architecture Specification*.
05h	Message Signaled Interrupts – This Capability structure identifies a device Function that can do message signaled interrupt delivery. Full documentation of this feature can be found in the *PCI Local Bus Specification*.
06h	CompactPCI Hot Swap – This Capability structure provides a standard interface to control and sense status within a device that supports Hot Swap insertion and extraction in a CompactPCI system. This Capability is documented in the *CompactPCI Hot Swap Specification PICMG 2.1, R1.0* available at http://www.picmg.org.
07h	PCI-X – Refer to the *PCI-X Protocol Addendum to the PCI Local Bus Specification* for details.

[2] The content of this chapter was originally in Appendix H of the *PCI Local Bus Specification, Revision 3.0*.

ID	Capability
08h	HyperTransport – This Capability structure provides control and status for devices that implement HyperTransport Technology links. For details, refer to the *HyperTransport I/O Link Specification* available at http://www.hypertransport.org.
09h	Vendor Specific – This Capability structure allows device vendors to use the Capability mechanism to expose vendor-specific registers. The byte immediately following the Next Pointer in the Capability structure is defined to be a length field. This length field provides the number of bytes in the Capability structure (including the Capability ID and Next Pointer bytes). All remaining bytes in the capability structure are vendor-specific.
0Ah	Debug port
0Bh	CompactPCI central resource control – Definition of this Capability can be found in the *PICMG 2.13 Specification* (http://www.picmg.com).
0Ch	PCI Hot-Plug – This Capability ID indicates that the associated device conforms to the Standard Hot-Plug Controller model.
0Dh	PCI Bridge Subsystem Vendor ID
0Eh	AGP 8x
0Fh	Secure Device
10h	PCI Express
11h	MSI-X – This Capability ID identifies an optional extension to the basic MSI functionality.
12h	Serial ATA Data/Index Configuration
13h	Advanced Features (AF) – Full documentation of this feature can be found in the *Advanced Capabilities for Conventional PCI ECN*.
14h	Enhanced Allocation
15h	Flattening Portal Bridge
Others	Reserved

3

3. Extended Capability IDs

This chapter describes the current Extended Capability IDs. Each Extended Capability structure must have an Extended Capability ID assigned by the PCI-SIG. Unless otherwise noted, each Extended Capability ID is defined in the *PCI Express Base Specification*. Companies wishing to define a new encoding should contact the PCI-SIG. All unspecified values are reserved for PCI-SIG assignment.

Table 3-1: Extended Capability IDs

ID	Extended Capability
0000h	Null Capability – This capability contains no registers other than those in the Extended Capability Header. It may be present in any Function. Functions may contain multiple instances of this capability. The Null Extended Capability is 32 bits and contains only an Extended Capability Header. The Capability Version field of a Null Extended Capability is not meaningful and may contain any value.
0001h	Advanced Error Reporting (AER)
0002h	Virtual Channel (VC) – used if an MFVC Extended Cap structure is not present in the device
0003h	Device Serial Number
0004h	Power Budgeting
0005h	Root Complex Link Declaration
0006h	Root Complex Internal Link Control
0007h	Root Complex Event Collector Endpoint Association
0008h	Multi-Function Virtual Channel (MFVC)
0009h	Virtual Channel (VC) – used if an MFVC Extended Cap structure is present in the device
000Ah	Root Complex Register Block (RCRB) Header
000Bh	Vendor-Specific *Extended Capability* (VSEC)
000Ch	Configuration Access Correlation (CAC) – defined by the *Trusted Configuration Space (TCS) for PCI Express ECN*, which is no longer supported
000Dh	Access Control Services (ACS)
000Eh	Alternative Routing-ID Interpretation (ARI)
000Fh	Address Translation Services (ATS)
0010h	Single Root I/O Virtualization (SR-IOV)

ID	Extended Capability
0011h	Multi-Root I/O Virtualization (MR-IOV) – defined in the *Multi-Root I/O Virtualization and Sharing Specification*
0012h	Multicast
0013h	Page Request Interface (PRI)
0014h	Reserved for AMD
0015h	Resizable BAR
0016h	Dynamic Power Allocation (DPA)
0017h	TPH Requester
0018h	Latency Tolerance Reporting (LTR)
0019h	Secondary PCI Express
001Ah	Protocol Multiplexing (PMUX)
001Bh	Process Address Space ID (PASID)
001Ch	LN Requester (LNR)
001Dh	Downstream Port Containment (DPC)
001Eh	L1 PM Substates
001Fh	Precision Time Measurement (PTM)
0020h	PCI Express over M-PHY (M-PCIe)
0021h	FRS Queueing
0022h	Readiness Time Reporting
0023h	Designated Vendor-Specific Extended Capability
0024h	VF Resizable BAR
0025h	Data Link Feature
0026h	Physical Layer 16.0 GT/s
0027h	Lane Margining at the Receiver
0028h	Hierarchy ID
0029h	Native PCIe Enclosure Management (NPEM)
002Ah	Physical Layer 32.0 GT/s
002Bh	Alternate Protocol
002Ch	System Firmware Intermediary (SFI)
Others	Reserved